Journal of
Theoretical and Applied Electronic Commerce Research

MDPI

Article

Preventing Return Fraud in Reverse Logistics—A Case Study of ESPRES Solution by Ethereum

Dong-Her Shih [1,*], Feng-Chuan Huang [1], Chia-Yi Chieh [1], Ming-Hung Shih [2] and Ting-Wei Wu [1]

[1] Department of Information Management, National Yunlin University of Science and Technology, Douliu 64002, Taiwan; D10323004@yuntech.edu.tw (F.-C.H.); edith850308@gmail.com (C.-Y.C.); portraits1129@gmail.com (T.-W.W.)

[2] Department of Electrical and Computer Engineering, Iowa State University, 2520 Osborn Drive, Ames, IA 50011, USA; mshih@iastate.edu

* Correspondence: shihdh@yuntech.edu.tw

Abstract: With the rapid development of e-commerce services, online retail has evolved from multi-channel to omni-channel in order to provide customers with more services. However, reverse logistics services (returns and exchanges) have become the target of many fraudulent activities, causing a lot of economic losses for many online retail companies. The current challenge of the traditional countermeasure is it requires a lot of manpower and training resources. In this study, we propose ESPRES, a system that adopts blockchain technology to prevent fraudulent behavior in the process of returns and exchanges with the smart contract and multi-attribute decision-support method to help consumers choose a suitable payment program. A practical implication of this study is that by adopting blockchain technology, a great amount of manpower used on determining whether each return or exchange is fraudulent can be reduced since merchants can check the product ownership. In addition, due to the fact that the footprint of goods cannot be forged, it can also prevent counterfeit or parallel imports of goods.

Keywords: reverse logistics; return fraud prevention; blockchain; ethereum; e-commerce; omnichannel logistics

check for updates

Citation: Shih, D.-H.; Huang, F.-C.; Chieh, C.-Y.; Shih, M.-H.; Wu, T.-W. Preventing Return Fraud in Reverse Logistics—A Case Study of ESPRES Solution by Ethereum. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 2170–2191. https://doi.org/10.3390/jtaer16060121

Academic Editors: Danny C.K. Ho and Tsz Leung Yip

Received: 2 July 2021
Accepted: 30 August 2021
Published: 1 September 2021

1. Introduction

With the development of logistics technology and the rise of e-commerce, consumers can quickly obtain the goods they need. A large number of goods are purchased, circulated, transported, and carried out at home and abroad. These activities are closely related to each other. The management of logistics is no longer just traditional logistics activities. Many companies have begun to integrate mobile platforms, e-commerce platforms, physical stores, and social network services. Omni-Channel Retailing [1] through the integration of online virtual and offline entities can help consumers purchase the goods they need more quickly. The omni-channel world is not only broadening the scope of channels, but also integrating the consideration of customer-brand-retail channel interactions, revenue impact into customer acquisition, frequency of orders, returns, and exchanges.

After the consumer has made a purchase, the product may need to be returned or exchanged due to a defect or mismatched description of the product. The process of returning the product from the consumer to the enterprise is called reverse logistics. As omni-channel retailing develops over time, fraudulent behaviors related to reverse logistics services have also increased, which is commonly called return/exchange fraud. Commodity return fraud is a growing problem, causing companies to lose millions of dollars each year [2].

Return fraud refers to the behavior of consumers returning goods to retailers, knowing that the return violates company or legal regulations (including returning functional but used or damaged goods). According to Parisi [3], the probability of exchange fraud using fake or stolen goods is about 40% on mobile phones and 90% on large household appliances. Return fraud caused at least $220 million in losses. Research on retailers also shows that 82% of mass-market retailers are aware of the return fraud problem [4]. Another research study on clothing returns shows that 50% of all returns are fraudulent [5]. In North America [6], approximately 8% of returns are fraudulent and the retailers' return policies are ineffectual. The traditional method of identifying return fraud is to manually review the requests with the help of blacklisting to prevent future frauds [2]. Unfortunately, this method has many problems and is limited by manpower, training, and expansion capabilities. It is worth noting that most sophisticated return scams can surely make a dent in retailers' profits.

The increasing amount of return fraud puts retailers in a challenging position. If they make the return policy strict, they may lose customers due to unpleasant return experiences. On the other hand, a liberal return policy is vulnerable to return abuse. The key challenge here is to form an idea of who your users are before shipping the item in the first place, which is only feasible with digital footprint analysis. Digital footprint analysis is a term used in fraud prevention. According to Zhuravlev et al. [7], due to the huge need for computational power on processing information, complex digital footprint analysis is only suitable for large retailers.

Nevertheless, several innovative solutions have made it possible for anyone to use anti-fraud tools in recent years. For instance, anti-fraud through image recognition, which takes a lot of computing power and time [8], or graph databases for financial fraud detection [9]. del Mar Roldán-García et al. [10] proposed an ontology-driven method for semantic conflict detection and classification. Rule-based expert systems are used to counter fraud in an e-commerce environment. A quick reverse email lookup check could tell a lot about a new buyer's return fraud risk based only on their email address. Similar checks could also be performed by gathering buyers' phone numbers during checkout. However, all these digital footprints only analyze the buyer's return fraud risk. They cannot prevent footprints from forgery. Research has shown that the introduction of blockchain in the supply chain is a good solution to return fraud prevention. Toyoda et al. [11] proposed a product ownership management system (POMS) of RFID-attached products for anti-counterfeits that can be used in the post-supply chain. This is a primary type of blockchain used in the post-supply chain which only proves the ownership of the product. However, they did not describe the complex detail in the context of reverse logistics, nor the interaction with other sales roles. Considering online store attributes which may be of importance to consumers during the purchase decision, the research found factors associated with shipping such as shipping fees, shipping speed, and return policy, to be essential decision criteria [12]. Therefore, we were motivated to provide a sophisticated solution to return fraud prevention and a freight fee decision support function in the issue of freight disputes with blockchain in reverse logistics.

Inspired by the past research [11,12], this study leverages the non-tamperable feature of the blockchain to prevent possible return fraud in retail and adds an extra function in the smart contracts of blockchain to help customers in freight fee decision support, a novel approach in current blockchain research. Digital footprints are protected from forgery by writing smart contracts on the Ethereum blockchain. The rights and interests of consumers, retailers, and others are also protected from reverse logistics activities.

In the following sections, we present preliminary knowledge and a literature review in Section 2, followed by the experimental design, proposed system flow, and the designed contracts (return contract, exchange contract, and management contract) in Section 3. Section 4 analyzes the common vulnerabilities of smart contracts. Section 5 will discuss our framework's ability, and Section 6 concludes our study with future works.

2. Preliminary

2.1. Reverse Logistics

The scope of reverse logistics is quite extensive. Thus, we focus on the management of reverse logistics activities that consumers return and exchange after products are sold to consumers.

According to Vitasek [13], reverse logistics is defined as the logistics activities through source reduction, recycling, substitution, reuse, disposal, etc. The Reverse Logistics Executive Council (RLEC) defines reverse logistics as the process of moving a product from its destination to another location. It is mainly to obtain the value that cannot be obtained in any other way, or to perform appropriate product disposal.

Fleischmann [14] defines reverse logistics as the process of planning, implementing, and controlling the efficiency of stored secondary products, and is opposed to the direction of the general supply chain to restore the value of its goods and more appropriate disposal. Rubio and Jiménez-Parra [15] believe that there are several reasons for implementing and planning reverse logistics:

1. Economy: The most direct reason is to reduce the use of raw materials and the cost of disposal, and to create added value for the final product. The indirect reason is demonstrating environmentally friendly and responsible behaviors to promote customer relations.
2. Law: In many countries such as the European Union, companies must be responsible for the recycling and related disposal of waste generated by the products produced or distributed by companies.
3. Society: Society realizes the importance of protecting the environment and the concept of sustainability, which leads to the companies' responsibility such as controlling carbon emissions and waste disposal.

As depicted in [16], the process of reverse logistics includes not only the return, exchange, or maintenance and upgrade services of general merchandise, but also the recycling process of converting merchandise into raw materials and scrap processing procedures. Fraud is part of the entire reverse logistics activity.

2.2. Return/Exchange Fraud

Return fraud is a fraudulent behavior using the return and exchange mechanism of goods. Common fraud methods can be divided into two types [3,4]. The first type is exchange fraud from buyers, in which stolen goods or fakes are exchanged for brand new products. In recent observations, the probability of exchange fraud is about 40% of mobile phones and up to 90% of large household appliances. The second type is unpacking return fraud, in which the valuable parts of the goods are removed and then returned for various reasons. Many malicious middlemen will use this method to resell the parts to other retailers, causing a large amount of financial loss for merchants.

Many service dynamics frameworks assume that consumers will not deliberately disrupt service contacts, but more and more studies believe that dysfunctional customer behavior is not uncommon. Wilkes [17] found that 98.6% of consumers believe that fraudulent returns are the most condemned consumer behavior. King [18,19] found that 82% of large retailers considered fraudulent returns as a major problem, and the fraudulent returns would reduce retail profits by 10–20%. In 2019 [20], the biggest European scam ever recorded by the National Retail Federation cost Amazon $370 K after a Spanish buyer stole items and returned boxes filled with dirt. The Appriss Retail report found the returns of online purchases were worth $41 Billion in total, where 35% (about $14 billion) were return frauds. In these respects, fraudulent returns by consumers have become a growing concern for contemporary retailers.

2.3. Blockchain Technology

Blockchain was developed as the core technology of Bitcoin. After about ten years of development, it has gradually become one of today's most breakthrough technologies, covering many industries such as finance, manufacturing, and educational institutions [21]. Blockchain is not just a single technology as it includes cryptography, mathematics, algorithms, and economic models, combined with peer-to-peer networks and distributed consensus algorithms to solve the problem of distributed database synchronization [22].

1. Bitcoin: Nakamoto [23] explained Bitcoin's mathematical logic, basic technical concepts, and how to use P2P networks to create electronic transaction systems that do not require dependence and trust. Bitcoin uses P2P architecture and cryptography principles to maintain the security of the entire Bitcoin network. P2P networks do not have a main server to operate. Participants of Bitcoin are user-end nodes with two roles: user and miner. The user can send and execute Bitcoin transactions. The miner is responsible for calculating the proof of work, broadcasting the output block to other nodes for verification, and then getting the corresponding amount of Bitcoin as a reward. The block is composed of multiple transactions. The transaction will be collected by the miners, and the address of the next block will be calculated and verified using the proof of work to generate a new block.

As shown in Figure 1, the blocks use addresses to link each other to form a database system. If the proof of work is calculated and the block is verified by other nodes, it will be written into the database to prove and record a series of events.

2. Ethereum: Ethereum [24] is one of the widely used blockchain networks in which a currency called Ether (ETH) is in circulation. Smart Contracts in Ethereum can be freely developed and executed in the blockchain. Smart contracts must be executed in the Ethereum Virtual Machine (EVM) connected to the Ethereum node and written in Solidity language. The user passes the transaction to the Ethereum network in order to create a new contract, invoke the function of the contract, and transfer ether to the contract or other users. All transactions will be recorded in the public additional data structure of the blockchain. The order of transactions on the blockchain determines the state of each contract and the balance of each user. Unlike the Bitcoin network, there are two types of accounts on Ethereum: the Externally Owned Account (EOA) and the Contract Account (CA). The EOA is an account held by a user with information such as the address and account balance. The CA is an account attached to the contract, which contains the address and balance like the EOA. The CA must be created by the user through the EOA transaction creation.

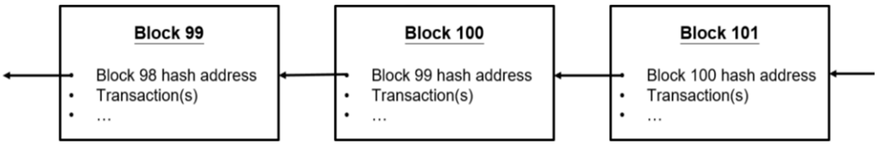


Figure 1. Link between the hash value and the blocks.

2.4. Research Questions

Reverse logistics services (returns and exchanges) have become the target of abuse or fraudulent activities which have caused a lot of economic losses for many online retail companies. Today's information systems in reverse logistics are usually developed based on the traditional logistics process with a third-party trust. On the other hand, most of the blockchain research is considering how to apply blockchain technology to real scenarios and reduce third-party regulation. For example, Sunny [25] discussed various blockchain-based traceability solutions, and many companies have gradually integrated related technologies. In IR (industrial revolution) 4.0, blockchain is considered to be a disruptive technology in the past decade, and the combination of blockchain with IR 4.0 technology is very important [26]. Public sector reforms are also affected by this blockchain trend, and e-government leaders around the world are tentatively beginning to grasp the potential of blockchain and other distributed ledger technologies [27]. Companies are increasingly using crowdsourcing platforms to bid for knowledge-intensive tasks in order to acquire scarce knowledge and skills that were otherwise unavailable. A blockchain-based reference architecture for knowledge-intensive crowdsourcing platforms was also proposed [28]. So, it is natural to apply blockchain to the return and exchange scenario of large household appliances. As mentioned in Section 1, our research question is how to use blockchain technology to prevent return fraud in large household appliances, and how to deploy a freight fee decision support function in smart contracts?

2.5. Large Household Appliances

Products that are likely to cause return fraud are usually mobile phones or large household appliances, which cause a lot of money and manpower losses [3]. It is difficult for general retailers to invest large amounts of human resources to check fraudulent activities; on the other hand, the policy of returning and replacing products cannot be tightened due to insufficient resources. To address these difficulties, this study aims to integrate blockchain technology and smart contracts and develop an ESPRES (**E**thereum **S**olution to **P**revent return fraud of large household appliances in **RE**verse logistic**S**) system to prevent return and exchange fraud in simulation. The solution is applicable to a retailer's process of selling large household appliances. Furthermore, with similar return and exchange procedures, our proposed ESPRES system can be adapted as the standard process for general large appliance vendors.

3. Proposed ESPRES System

3.1. Delivery and Return Scenarios of Large Home Appliances

Using blockchain technology for the system is not always a good option as various factors of the system have to be considered. To determine whether the blockchain technology is suitable for the return and exchange situation of large household appliances, a recommended process is proposed. The first step is to determine the role of the situation, then the trust relationship and interaction between the roles must be determined. Finally, the results collected in the above steps are combined to arrive at a draft architecture [29].

The delivery and return scenarios of large home appliances are shown in Figure 2. Steps one to four are the current process of delivering goods to consumers in the supply chain, and steps five to eight are the return and exchange of goods. The detailed steps are as follows:

1. The product is delivered by the manufacturer to the retailer.
2. After the manufacturer sends the large appliances to the retailer, the retailer sells them to the consumer.
3. After consumers purchase goods from retailers, they are distributed by logistics companies and delivered to designated locations.
4. After the large home appliances are delivered to the designated location, maintenance personnel will install them.
5. During the re-appreciation period, if the consumer finds that the purchased product is faulty or inconsistent with the original, the consumer can perform the return and exchange procedure through the e-commerce platform.
6. After the consumer applies for a return, the retailer will notify the maintenance staff to go to the designated place for disassembly.
7. After receiving the notice, the maintenance will go to the designated place for disassembly procedures.
8. After the dismantling of the goods is completed, the logistics company will take it back to the designated location.
9. The logistics company ships the large appliances back to retailers.
10. The retailer sends the goods returned by the consumer back to the manufacturer for testing.

If consumers engage in fraudulent returns and exchanges through fake invoices, stolen goods, or other goods, the merchant often will not find the problem until the entire reverse logistics activity is finally returned to the original manufacturer. Even if manufacturers set clear regulations to prevent return fraud, such as requiring consumers to have an invoice in order to proceed with the subsequent refund process [30], it cannot prevent forgery of invoices or theft of others' invoices. Therefore, when the entire return fraud occurs, the loss is not only the monetary value of the product itself, but also the waste of manpower, logistics activities, and time resources.

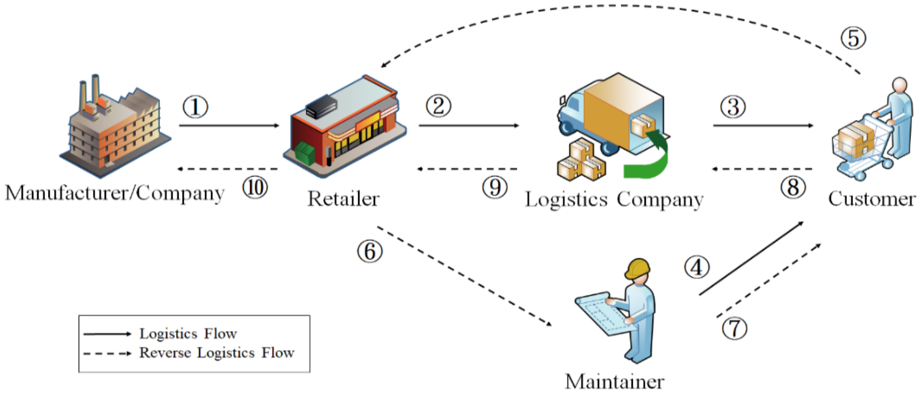


Figure 2. Large home appliances delivery and return scenario.

3.2. ESPRES System

Blockchain is not a trustless technology but rather a confidence machine. The concept of trust is to reduce the complexity of the environment through the relationship of trust [31]. The proposed Ethereum solution of the return fraud prevention system, ESPRES, is shown in Figure 3.

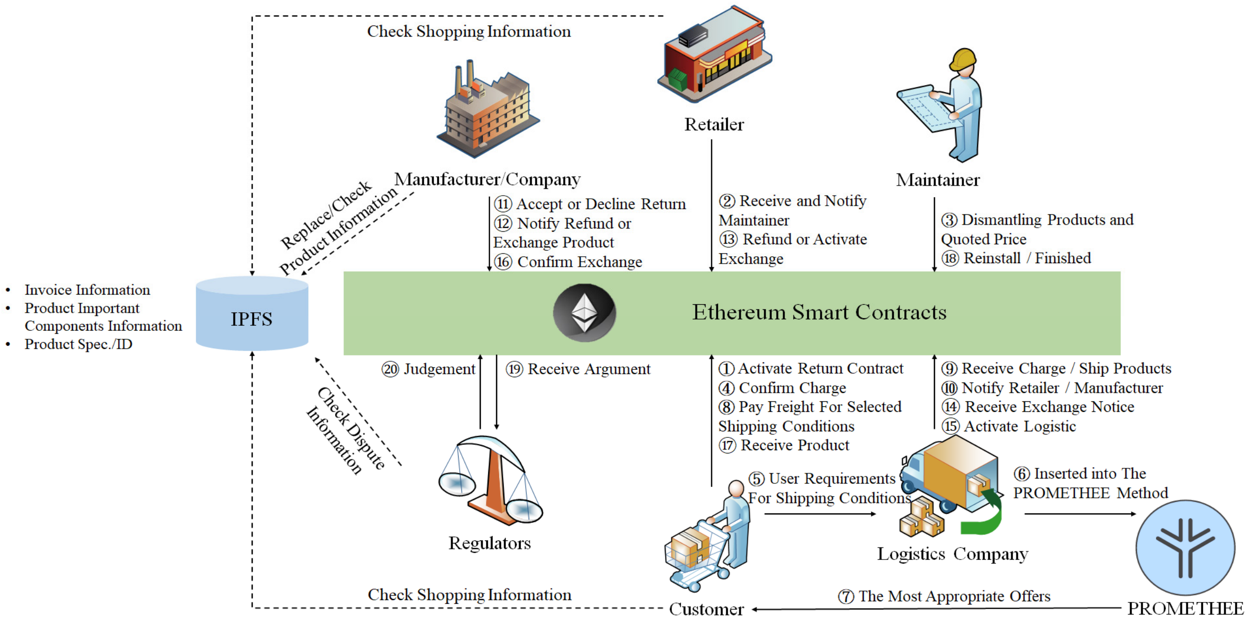

Check Shopping Information
Retailer
Manufacturer/Company
Maintainer
Replace/Check Product Information
⑪ Accept or Decline Return
⑫ Notify Refund or Exchange Product
⑯ Confirm Exchange
② Receive and Notify Maintainer
⑬ Refund or Activate Exchange
③ Dismantling Products and Quoted Price
⑱ Reinstall / Finished
• Invoice Information
• Product Important Components Information
• Product Spec./ID
IPFS
Ethereum Smart Contracts
⑳ Judgement
⑲ Receive Argument
① Activate Return Contract
④ Confirm Charge
⑧ Pay Freight For Selected Shipping Conditions
⑰ Receive Product
⑨ Receive Charge / Ship Products
⑩ Notify Retailer / Manufacturer
⑭ Receive Exchange Notice
⑮ Activate Logistic
Check Dispute Information
Regulators
⑤ User Requirements For Shipping Conditions
Logistics Company
⑥ Inserted into The PROMETHEE Method
Check Shopping Information
Customer
⑦ The Most Appropriate Offers
PROMETHEE

Figure 3. The proposed ESPRES system for return fraud prevention.

In the process of writing a smart contract, it is necessary to clearly define which role has the authority to execute the contract. Not every role can use all contracts in the blockchain network. Therefore, this section establishes roles and available smart contracts based on the return and exchange scenarios in Figure 2. Each participating role has an Ethereum address (EA) and participates through the functions in the smart contract. The Ethereum return and exchange system flow is described as follows:

1. When customers are dissatisfied with the products or have defects or malfunctions during the appreciation period, they will request a return or exchange, which activates the return smart contract.
2. After receiving the customer's return application, the retailer will notify the maintainer.
3. The maintainer will go to the place designated by the customer for disassembly, assembly, and to quote the cost to the customer.
4. After the customer confirms the quotation for the disassembly and assembly costs of the maintainer, the payment is made.
5. The customer makes requests for delivery, such as delivery time, logistics company, etc.
6. The logistics company incorporates the delivery requirements and puts the customer's requirement into the PROMETHEE [32] decision.
7. PROMETHEE sorts and provides customers with the most suitable shipping method and freight.
8. Customers confirm and pay the freight to the logistics company.
9. The logistics company sends the large household electrical appliances back to the manufacturer at the place designated by the customer.
10. The logistics company informs the retailer and the manufacturer.
11. After the manufacturer receives the returned goods and conducts inspections, they will decide whether to accept the returned goods or not.
12. If the manufacturer accepts, it will notify the retailer to refund the customer. If the manufacturer accepts the exchange, it will notify the retailer to proceed with the following replacement procedure.
13. If the retailer receives the manufacturer's refund notification, it will refund the money to the customer; if it receives the manufacturer's replacement notification, the replacement procedure will be initiated.
14. If the retailer initiates the replacement procedure, the logistics company will receive the replacement notification.
15. The logistics company starts the logistics to deliver the goods, and collects the goods to be replaced or repaired to the manufacturer.
16. The manufacturer confirms the exchange, pays the freight to the logistics company, and the logistics company will deliver the goods to customers.
17. If the customer receives the replaced product, they will notify the maintainer.
18. After receiving the notice from the customer, the maintainer will reinstall the large household appliances at the place designated by the customer, and the transaction will end.
19. When there are disputes in the process of return and exchange, the regulators will evaluate and judge.
20. The judged result will be transmitted back to the party who brought the dispute.
21. Relevant information will be backed up to the Inter Planetary File System (IPFS). IPFS is a network transmission protocol designed to establish persistent and distributed storage and sharing of files.

Figures 4–6 are system diagrams for this study. The blue line represents the function in the smart contract, which will be executed after obtaining the corresponding conditions or information. The red lines are real-world events, representing the interaction between customers, products, and employees. Figure 4 outlines the sequence flow of the customer executing the function ActivateReturnContract() to start the smart contract. Figure 5 shows that the logistics company delivers goods to the manufacturer and notifies the retailer and the manufacturer, and calls the event ShipProductsAndNotify (Logistics Company EA, Retailer EA, Manufacturer/Company EA, Product Owner). Figure 6 shows the manufacturer confirms the replacement procedure, pays the freight and transfers the ownership of the goods to the logistics company, and calls the event ConfirmExchange (Manufacturer/Company EA, Logistics Company, Freight, Product Owner). An entity-relationship diagram of our proposed ESPRES system is also shown in Figure 7.

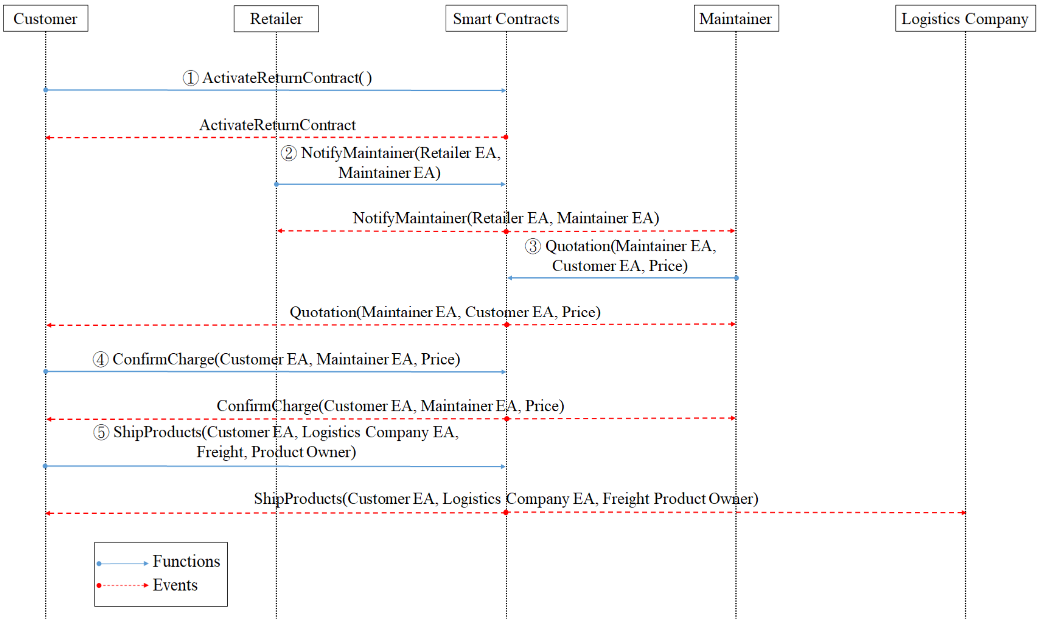

Customer
Retailer
Smart Contracts
Maintainer
Logistics Company
① ActivateReturnContract( )
ActivateReturnContract
② NotifyMaintainer(Retailer EA, Maintainer EA)
NotifyMaintainer(Retailer EA, Maintainer EA)
③ Quotation(Maintainer EA, Customer EA, Price)
Quotation(Maintainer EA, Customer EA, Price)
④ ConfirmCharge(Customer EA, Maintainer EA, Price)
ConfirmCharge(Customer EA, Maintainer EA, Price)
⑤ ShipProducts(Customer EA, Logistics Company EA, Freight, Product Owner)
ShipProducts(Customer EA, Logistics Company EA, Freight Product Owner)
Functions
Events

Figure 4. System Sequence Diagram 1.

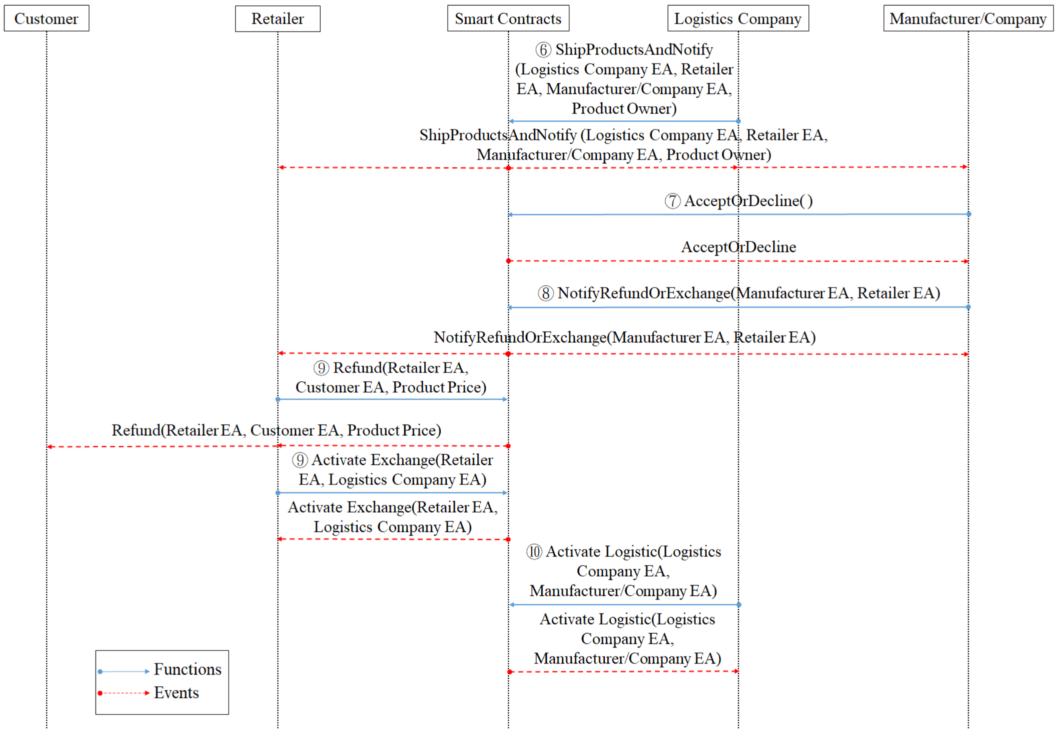

Customer
Retailer
Smart Contracts
Logistics Company
Manufacturer/Company
⑥ ShipProductsAndNotify (Logistics Company EA, Retailer EA, Manufacturer/Company EA, Product Owner)
ShipProductsAndNotify (Logistics Company EA, Retailer EA, Manufacturer/Company EA, Product Owner)
⑦ AcceptOrDecline( )
AcceptOrDecline
⑧ NotifyRefundOrExchange(Manufacturer EA, Retailer EA)
NotifyRefundOrExchange(Manufacturer EA, Retailer EA)
⑨ Refund(Retailer EA, Customer EA, Product Price)
Refund(Retailer EA, Customer EA, Product Price)
⑨ Activate Exchange(Retailer EA, Logistics Company EA)
Activate Exchange(Retailer EA, Logistics Company EA)
⑩ Activate Logistic(Logistics Company EA, Manufacturer/Company EA)
Activate Logistic(Logistics Company EA, Manufacturer/Company EA)
Functions
Events

Figure 5. System Sequence Diagram 2.

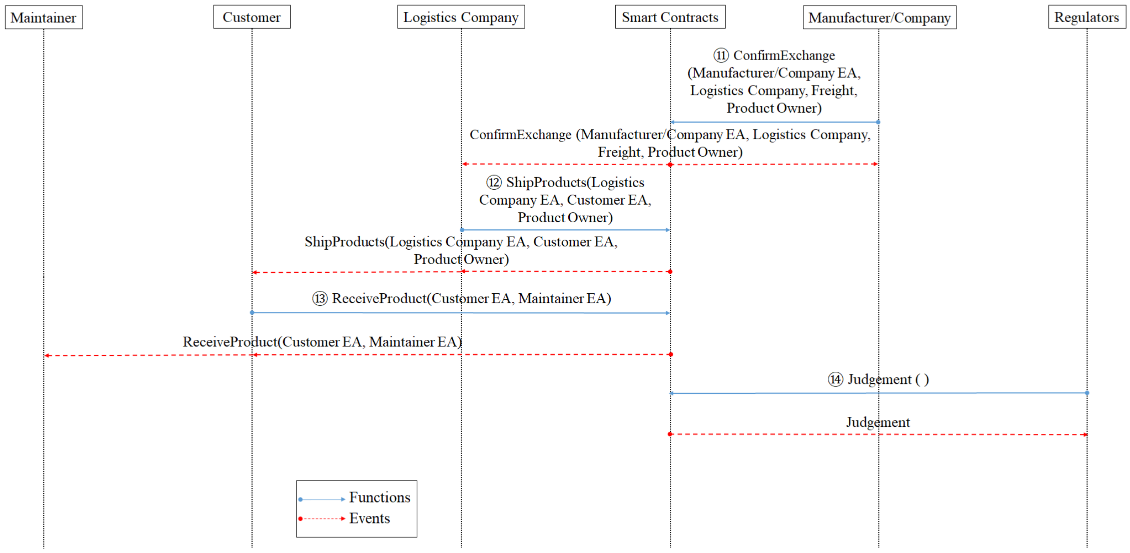

Maintainer
Customer
Logistics Company
Smart Contracts
Manufacturer/Company
Regulators
⑪ ConfirmExchange (Manufacturer/Company EA, Logistics Company, Freight, Product Owner)
ConfirmExchange (Manufacturer/Company EA, Logistics Company, Freight, Product Owner)
⑫ ShipProducts(Logistics Company EA, Customer EA, Product Owner)
ShipProducts(Logistics Company EA, Customer EA, Product Owner)
⑬ ReceiveProduct(Customer EA, Maintainer EA)
ReceiveProduct(Customer EA, Maintainer EA)
⑭ Judgement ( )
Judgement
Functions
Events

Figure 6. System Sequence Diagram 3.

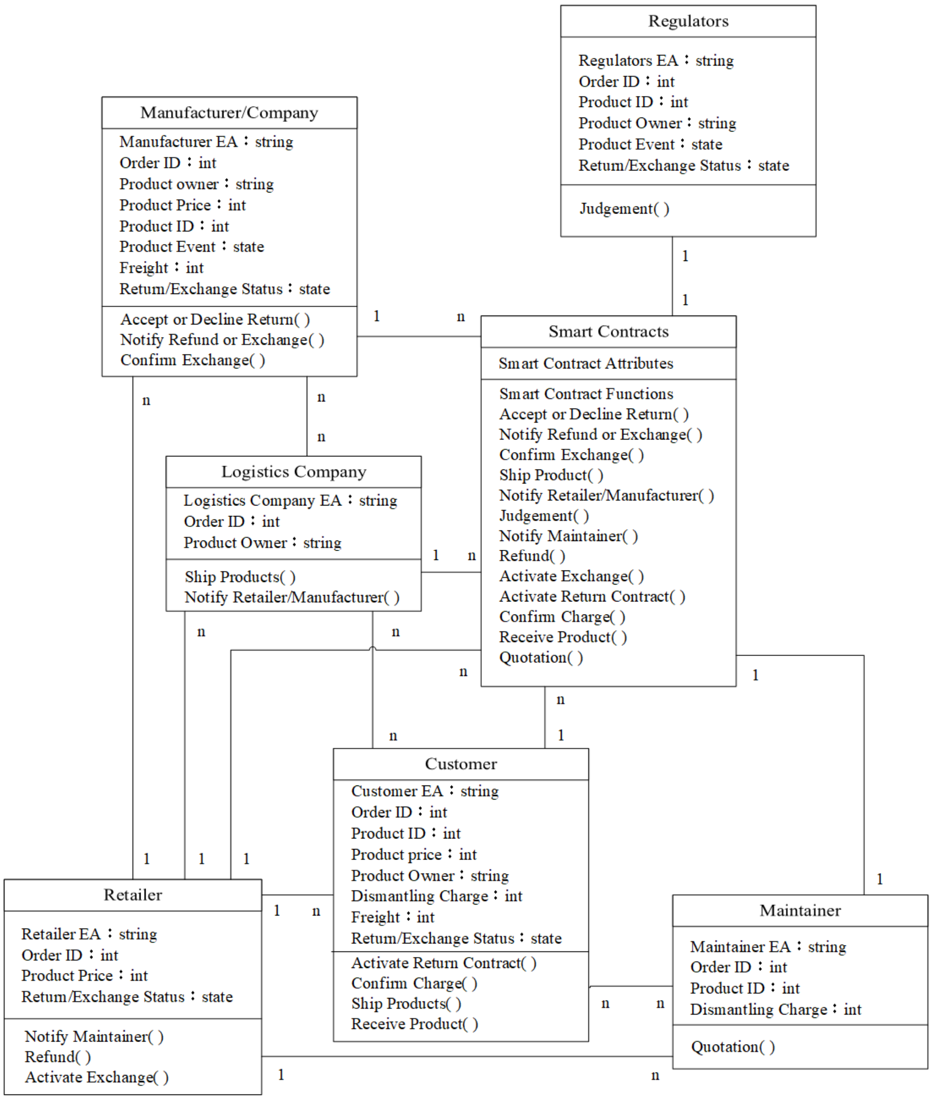


Figure 7. Entity-relationship diagram.

3.3. Implemented Contracts

Smart contract technology is reshaping traditional industries and business processes. The smart contract embedded in the blockchain can automatically execute the contract terms of the agreement without the intervention of a trusted third party. Smart contracts can reduce management and service costs, improve the efficiency of business processes and reduce risks [33]. The programming language for writing smart contracts in this research is Solidity, a contract object-oriented programming language, and Ethereum is designated as the programming language for development. All contract creation (development) and testing use Remix IDE. Remix IDE is an official Solidity online IDE, which provides testing and debugging functions and can be easily compiled.

In this research, we design three smart contracts: the return contract (RC), exchange contract (EC), and management contract (MC). The RC records the entire return process to this contract, including the customer's initiation of the return, receipt and disassembly costs and freight, and the manufacturer's decision to accept returns and the retailer's return of the payment. The EC records the exchange process, including the retailer's initiation of the exchange procedure and the transfer of product ownership. The MC can view the current status of all transactions, including the completion of disassembly and assembly, product delivery, refund completion, unpaid payment, return completion, exchange completion, etc.

(1) Algorithm 1 Activate: Start a return for customers, start an exchange for retailers, or start a logistics process for a logistics company.

(2) Algorithm 2 Notify: When the retailer informs the maintainer to disassemble and assemble large household appliances, the logistics company informs the retailer and the manufacturer to deliver the goods that the customer wants to return to the manufacturer; the manufacturer notifies the retailer to return after accepting the return. If the replacement is accepted, the retailer will be notified to perform the replacement procedure. When the customer receives the replaced product and notifies the maintainer for installation, the notification contract shall be used.

(3) Algorithm 3 Dismantling Charge Quoted Price: The maintainer visits the place designated by the customer to dismantle and assemble the large household appliances and quote the dismantling charge to the customer using the quotation contract.

(4) Algorithm 4 Freight Decision: Based on customers' personal preferences, the system will choose suitable transportation attributes (transport time, product volume, transportation distance), make decisions, provide customers with the most appropriate transportation plan, and multi-attribute freight. The function of the decision system is added to the contract.

(5) Algorithm 5 Payment (Charge): The customer pays the disassembly and assembly cost quoted by the maintainer, the return shipping fee to the logistics company, the retailer refunds the customer, and the manufacturer confirms the exchange and pays the freight to the logistics company. A payment contract is required.

(6) Algorithm 6 Transfer Product Owner: The ownership transfer contract is for the logistics company to transport the disassembled large household appliances from the customer's designated location to the manufacturer and transfer the ownership of the product. After the manufacturer confirms the exchange, the logistics company will re-ship from the manufacturer to the customer, and then transfer the ownership of the goods.

(7) Algorithm 7 Company Decision: After the manufacturer receives the goods returned by the customer, it chooses whether to accept the return or exchange after testing.

(8) Algorithm 8 Argument: When there is a dispute in the process of return and exchange, the supervisor will evaluate and make a ruling, and return the ruling result.

Algorithm 1 Activate

```
Input: Ethereumaddress(EA) of Customer
Ethereumaddress(EA) of Retailer
Ethereumaddress(EA) of Logistics Company
OrderNumber, ProductID, ProductOwner
Contractstate is Created
Restrict access to only Customer, Retailer, or Logistics Company
if Customer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Activated
            Create a notification message stating Return
        end
    end
else if Retailer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        Contract state changes to Activated
        Create a notification message stating Exchange
    end
else if Logistics Company access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            if ProductOwner = Manufacturer then
                Contract state changes to Activated
                Create a notification message stating Logistics
            end
        end
    end
else
    Revert contract state and show an error
end
```

Algorithm 2 Notify

```
Input: Ethereumaddress(EA) of Retailer
Ethereumaddress(EA) of Logistics Company
Ethereumaddress(EA) of Manufacturer
Ethereumaddress(EA) of Customer
Ethereumaddress(EA) of Maintainer
Notification
Restrict access to only Retailer, Logistics Company, Manufacturer, Customer
if Retailer access = agree then
    Create a notification message stating Maintainer Dismantling Products
else if Logistics Company access = agree then
    Create a notification message stating Shipping Product to Manufacturer
else if Manufacturer access = agree then
    if Return = accept then
        Create a notification message stating Retailer Refund
    else if Exchange = accept then
        Create a notification message stating Retailer Exchange
    end
else if Customer access = agree then
    Create a notification message stating Maintainer Reinstall Products
else
    Revert contract state and show an error
end
```

Algorithm 3 Dismantling Charge Quoted Price

```
Input: Ethereumaddress(EA) of Maintainer
Ethereumaddress(EA) of Customer
OrderNumber, ProductID, DismantlingCharge
Contractstate is Activated
Restrict access to only Maintainer
if Maintainer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Dismantled
            Create a notification message stating Dismantling Charge
        end
    end
else
    Revert contract state and show an error
end
```

Algorithm 4 Freight Decision

```
Input: Ethereumaddress(EA) of Logistics Company
Ethereumaddress(EA) of Customer
Contractstate is Activated
Restrict access to only Customer
if Customer access = agree then
    Plans_sorce = function decision(Logistics Companys,
    Transportation Time, Distance, Product Volume)
    print (Plans_sorce)
else
    Revert contract state and show an error
end
```

Algorithm 5 Charge

```
Input: Ethereumaddress(EA) of Customer
Ethereumaddress(EA) of Maintainer
Ethereumaddress(EA) of Logistics Company
Ethereumaddress(EA) of Retailer
Ethereumaddress(EA) of Manufacturer
OrderNumber, ProductID, DismantlingCharge, Freight, ProductPrice
Restrict access to only Customer, Retailer, Manufacturer
if Customer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            if DismantlingCharge = paid then
                Create a notification message stating Payment Dismantling Charge
            else if Freight = paid then
                Create a notification message stating Payment Freight
            end
        end
    end
else if Retailer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductPrice = paid then
            Create a notification message stating Refund
    end
else if Manufacturer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if Freight = paid then
            Create a notification message stating Payment Freight
        end
    end
else
    Revert contract state and show an error
end
```

Algorithm 6 Transfer Product Owner

```
Input: Ethereumaddress(EA) of Customer
Ethereumaddress(EA) of Logistics Company
Ethereumaddress(EA) of Manufacturer
OrderNumber, ProductID, ProductOwner
Contractstate is Dismantled
Restrict access to only Customer, Logistics Company, Manufacturer
if Customer access = agree and Freight = paid then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Transferred
            Transfer Product Owner to Logistics Company
            Create a notification message stating Transferred Product Owner
        end
    end
else if Logistics Company access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            if Customer Freight = received then
                Contract state changes to Transferred
                Transfer Product Owner to Manufacturer
                Create a notification message stating Transferred Product Owner
            else
                if Manufacturer Freight = received then
                    Contract state changes to Transferred
                    Transfer Product Owner to Customer
                    Create a notification message stating Transferred Product Owner
                else
                    Revert contract state and show an error
                end
            end
        end
    end
else if Manufacturer access = agree and Freight = paid then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Transferred
            Transfer Product Owner to Logistics Company
            Create a notification message stating Transferred Product Owner
        end
    end
else
    Revert contract state and show an error
end
```

Algorithm 7 Company Decision

```
Input: Ethereumaddress(EA) of Manufacturer
OrderNumber, ProductID
Contractstate is Transferred
Restrict access to only Manufacturer
if Manufacturer access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Decided
            Create a notification message stating Accepted or Declined Return
        end
    end
else
    Revert contract state and show an error
end
```

Algorithm 8 Argument

```
Input: Ethereumaddress(EA) of Regulator
OrderNumber, ProductID
Restrict access to only Regulator
if Regulator access = agree then
    if OrderNumber = 0 then
        str = OrderNumber no register!!
    else
        if ProductID = 0 then
            str = ProductID no register!!
        else
            Contract state changes to Judged
            Transfer the Product Contract balance to Regulator
        end
    end
else
    Revert contract state and show an error
end
```

4. Security Analysis and Comparison

Although smart contracts provide a certain degree of transparency, there are still some weaknesses in the design that should be considered. Past research [34] shows the security issues of smart contracts usually were Reentrancy Vulnerability, Transaction-Ordering Dependence, Mishandled Exceptions, and Timestamp Dependence as described below:

1. Reentrancy Vulnerability: When a smart contract uses functions related to remittances, it is possible that Reentrancy Vulnerability may be generated due to processing order issues in the design. In other words, if the remittance is performed before the storage status is changed, a malicious attacker will create a new contract through the loophole to steal the Ether in the victim contract. In June 2016, a German startup company DAO, was a victim of this loophole and a market value of approximately US$50 million was stolen. As long as the remittance process is involved, the issue of Reentrancy must be taken seriously. There is a design pattern called the checks-effects-interactions pattern that can solve this problem. First, in the Checks phase, the designer must determine whether the conditions are met, such as the use of the require() function. The second stage, Effects, is to update the state in the contract. In the final Interactions

stage, remittance instructions or exchange messages with other contracts or accounts are used.

2. Transaction-Ordering Dependence: Since the order in which the transactions are included in the block depends on the miners, sometimes the transactions cannot be executed in the planned order. This problem is called Transaction-Ordering Dependence (TOD).
3. Mishandled Exceptions: If an exception occurs when calling a function in Ethereum, the contract must be aborted and the recovery state returns false. However, the design may have abnormal behavior that will not be directly returned to the user.
4. Timestamp Dependency: In smart contract design," block.timestamp" or "now" is often used to obtain the timestamp of the block. When using these numbers for calculations, miners have a certain degree of ability to master the write time. For example: using block.timestamp to calculate a random number lottery, as long as a miner has the ability to mine a block at a specific time, it can be rewarded by participating in the lottery.
5. Cryptolojacking Lifecycle: One-third of the Cryptojacking samples disappear within 15 days with frequent updates [35].

4.1. Cost Analysis

Table 1 shows the address of all accounts in the ESPRES system. The account identities are manufacturer (M), retailer (R), logistics company (L), maintenance personnel (m), consumer (C), and supervisor (r). All subsequent tests will use the following six accounts to test smart contracts through Remix IDE.

Table 1. System account address.

Account	Address
M	0x7881A5C6014bC25C43458f41eBFD16249c87BdB9
R	0xc9e13c75d044B1F4b0bCDeB7a828705be5d958BD
L	0xcF8d3353A98140bf3A5e9E722C9eCCa1B01e4682
m	0x1D697fd294CAbE52396aD0afbF2dAcBfb3234Ad5
C	0x6B0614189C986e39298590E0eAF31707371d54ab
r	0x456Fb86e15AF1a226fAF106bF6716Af80F585336

When the transaction is executed in an Ethereum smart contract, the fuel required for consumption is called Gas. The unit price of Gas is called Gas Price, and the amount of Gas consumed multiplied by the Gas Price will be paid to miners as a handling fee. When issuing a Transaction, the Gas Limit parameter is used to set the upper limit of Gas usage to avoid the consumption of handling fees caused by the incorrect execution of programming errors. In this study, when the ESPRES system was tested, the Gas Limit was the default value of Remix IDE 3,000,000. Table 2 shows the cost of gas consumed by the functions used in the smart contract of the ESPRES system for testing this research. Using the data on the CoinGecko website on 22 June 2020, one unit of ETH is equivalent to US $238.36.

4.2. Security Analysis Report

SECURIFY [36,37] is a security scanner for Ethereum smart contracts, SECURIFY conducted an extensive evaluation of real-world Ethereum smart contracts and proved the correctness of the smart contract and the discovery of serious violations. This study uses Remix IDE to test the smart contract and our source codes written for the Return Contract, Exchange Contract, and Management Contract are sent to SECURIFY for testing (https://github.com/kk3329188/lib.git, accessed on 30 July 2021). Figure 8a–c are security analysis reports for our proposed Return and Exchange and Management contracts. Figure 8a shows that after the RC conducts a security check, the result shows that this contract can be written to the storage without restriction, which means that the part of the contract storage that all users can write to the sensitive field of the contract may be very dangerous. The contract fields that can be modified by any user must be checked. And this contract is vulnerable to the Locked Ether vulnerability attack because it allows users to receive but does not allow users to select or destroy contracts from it. Figure 8b shows the result of EC security detection. This contract is vulnerable to the Locked Ether vulnerability attack because it allows users to deposit Ether by calling the deposit function, and it does not contain any functions that allow users to choose to deposit Ether. Figure 8c shows the result of MC's security inspection.

Table 2. The cost of function Gas used in smart contract.

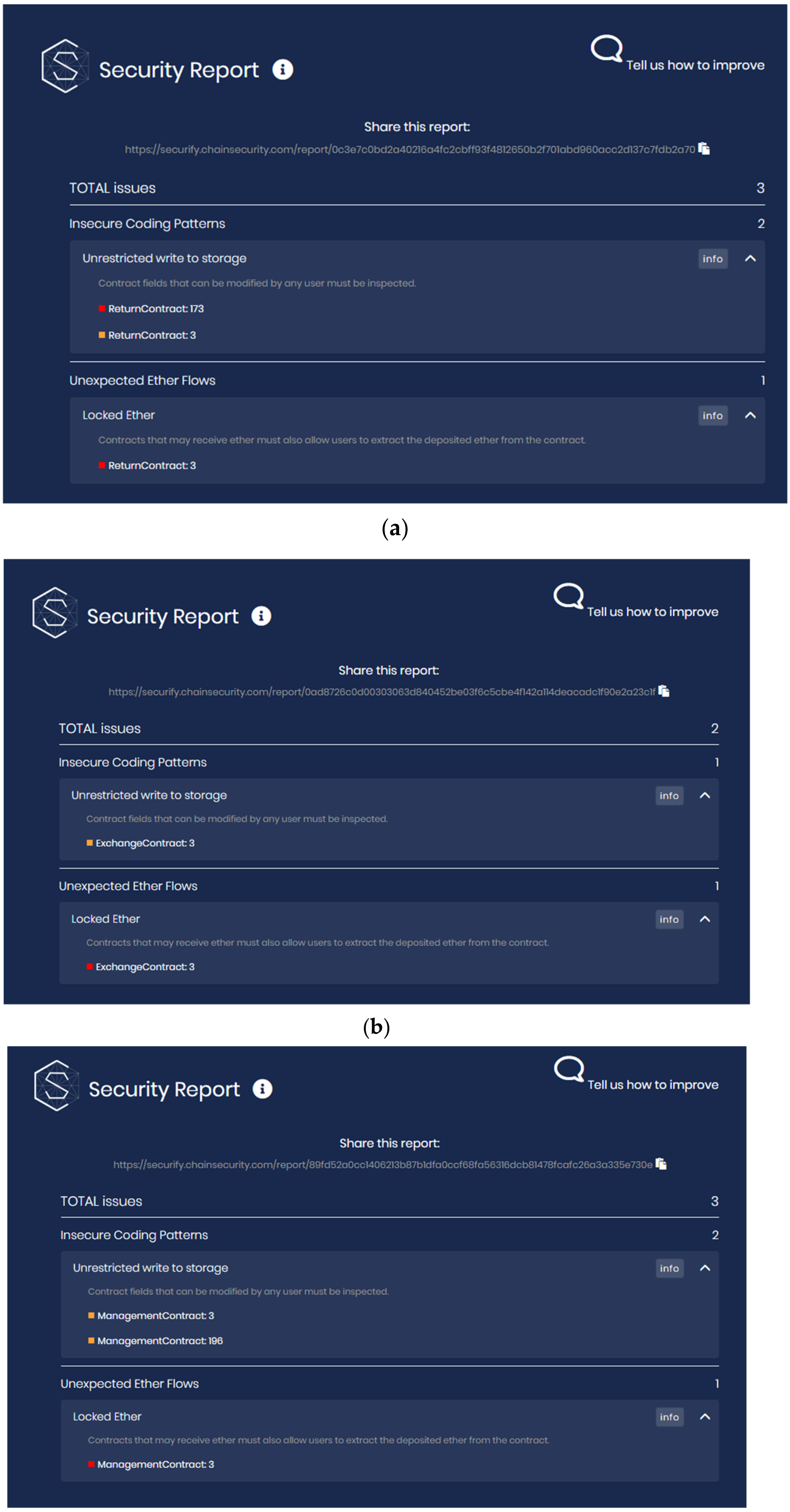
Function Name	Transaction Cost	Execution Cost	USD
activate()	47,675	24,227	0.0114
notify()	26,293	4638	0.0063
quotedprice()	27,332	5478	0.0065
charge()	163,627	141,734	0.0390
transferproductowner()	86,875	98,415	0.0207
companydecision()	27,428	5237	0.0065
argue()	93,425	85,859	0.0223

4.3. Comparison

In order to satisfy consumers in the environment of the supply chain, the return and exchange procedures must be kept unimpeded. Traditional centralized systems [38] cannot distinguish whether the consumers are malicious or not, and it is often necessary to fully accept the return and exchange procedures of the consumers. This can cost a lot in terms of disassembly and transportation. In addition, traditional systems must collect long-term historical information to determine possible fraudulent behaviors in transactions. Retailers must tighten the return and exchange process, such as reducing the number of days for return, having consumers share part of the cost, continuous telephone contacts, and so on.

The return and exchange activity are part of the supply chain. Many supply chain-related activities try to solve some of the problems through integration into the blockchain. It also represents that the current traditional supply chain system is facing huge challenges. Through the integration of blockchain technology, such as Hyperledger [39], Exonum [40], and Ethereum [41], system owners can set permissions for specific attributes to be visible to some characters, or set them as private attributes, which cannot be fully replicated even on public blockchain networks. In addition, the blockchain information cannot be tampered with and is independent, which can effectively prevent most frauds. Therefore, blockchain technology is necessary for the reverse logistics scenario in fraud prevention.

The relevant studies in preventing fraudulent activity on the supply chain are shown in Table 3. Most of the studies are aimed at the traceability of commodities or fraudulent activity that may occur in transactions, However, our study on large home appliances return/exchange can totally prevent fraudulent returns and exchanges by interested parties.



Security Report

Tell us how to improve

Share this report:

https://securify.chainsecurity.com/report/0c3e7c0bd2a40216a4fc2cbff93f4812650b2f701abd960acc2d137c7fdb2a70

TOTAL issues 3

Insecure Coding Patterns 2

Unrestricted write to storage

Contract fields that can be modified by any user must be inspected.

🟥 ReturnContract: 173

🟧 ReturnContract: 3

Unexpected Ether Flows 1

Locked Ether

Contracts that may receive ether must also allow users to extract the deposited ether from the contract.

🟥 ReturnContract: 3

(**a**)

Security Report

Tell us how to improve

Share this report:

https://securify.chainsecurity.com/report/0ad8726c0d00303063d840452be03f6c5cbe4f142a114deacadc1f90e2a23c1f

TOTAL issues 2

Insecure Coding Patterns 1

Unrestricted write to storage

Contract fields that can be modified by any user must be inspected.

🟧 ExchangeContract: 3

Unexpected Ether Flows 1

Locked Ether

Contracts that may receive ether must also allow users to extract the deposited ether from the contract.

🟥 ExchangeContract: 3

(**b**)

Security Report

Tell us how to improve

Share this report:

https://securify.chainsecurity.com/report/89fd52a0cc1406213b87b1dfa0ccf68fa56316dcb81478fcafc26a3a335e730e

TOTAL issues 3

Insecure Coding Patterns 2

Unrestricted write to storage

Contract fields that can be modified by any user must be inspected.

🟧 ManagementContract: 3

🟧 ManagementContract: 196

Unexpected Ether Flows 1

Locked Ether

Contracts that may receive ether must also allow users to extract the deposited ether from the contract.

🟥 ManagementContract: 3

(**c**)

Figure 8. (**a**) Security analysis report for the Return contract. (**b**) Security analysis report for the Exchange contract. (**c**) Security analysis report for the Management contract.

Table 3. Comparison of other studies.

Paper	Platform	Identity Verification	Transaction Fraud Prevention	Return Fraud Prevention
[11]	Ethereum	Yes	Yes	Partially
[35]	Web		Yes	-
[39]	Hyperledger	Yes	Yes	-
[40]	Exonum	Yes	Yes	-
[41]	Ethereum	Yes	Yes	-
Ours	Ethereum	Yes	Yes	Yes

5. Framework Discussion

Blockchain systems are very effective in preventing objective information fraud, such as loan application fraud, where fraudulent information is fact-based [42]. The typical organization loses 5% of its revenues to fraud each year, according to a study by the Association of Certified Fraud Examiners. Unfortunately, fraud in a business can go undetected for a long time and is often hard to uncover. The three features of blockchains are that they are, distributed, immutable, and can be permissioned, making business networks less susceptible to fraud.

The proposed ESPRES System was built on top of a permissionless blockchain technology, which benefits from the transparency and immutability of the data. Transactions between users are provided using smart contracts. Smart contracts eliminate third-party contacts and all transactions with the decentralized application created are recorded on the blockchain network. Recorded transactions are maintained in accordance with the principles of confidentiality, integrity, and availability through blockchain and smart contracts. In the application developed with the use of smart contracts, the user can access information such as location, price list, payment type, etc. for freight fee decisions by using PROMETHEE [32]. PROMETHEE, a multi-criteria decision-making method, is used in case there is more than one offer suitable for a user request and it is recorded in the blockchain to avoid possible future freight fee disputes. PROMETHEE has been adopted in other areas such as Electrical Vehicle Charging Platform decisions [43] and the most suitable type of Cryptocurrency for investment [44]. With the help of smart contracts, the established parameters are incorporated into the contract, and the contract will be executed as long as the corresponding conditions are met. The ESPRES System needs to be registered first to facilitate identity verification. After identity verification, other functions can be used, such as registered consumers, the return contract can only be activated after purchasing the merchandise, and no other malicious user can arbitrarily activate the return contract to which other consumers belong.

According to the current framework, there may be several malicious return fraud possibilities:

1. Consumers perform return fraud through fake invoices.
2. Consumers try to execute return fraud through parallel imports.
3. Consumers try to collude with maintenance workers for return and exchange fraud.

In the first two situations, the retailer will look for the consumer’s purchase record, invoice information, product manufacturing information, etc., based on the block for verification. Once a problem is found in the verification process, it does not cost the disassembly and transportation costs. Instead, consumers are required to submit real information and verify the follow-up process. In the third case, the consumer submitted real information to the retailer for disassembly procedures and colluded with the maintenance worker to conduct return and exchange fraud. Logistics companies will still perform information verification when delivering goods on-site to prevent parallel imports or merchandise with unknown product information from being transferred to the manufacturer. Through smart contracts, many fraud situations can be prevented and unnecessary costs can be reduced.

In the post-COVID-19 period, the global supply chain also has major problems. Cross-industry shutdowns, long-distance work at home, reduction of actual meetings, etc., have led to delays in the delivery of goods in the global supply chain [45]. During this period, research on the supply chain with innovation and technology has gradually increased. Singh et al. [46] proposed a scenario similar to COVID-19, by using drones provided by suppliers, smart contracts are created through blockchain solutions to build the trust of buyers and sellers, or combine 5G and Internet of Things technology to help retail, medical and other businesses [47]. Kumar et al. [48] believe that the challenges faced by the retail industry can build trust and transparency through Industry 4.0 technology, and can effectively manage medical necessities. Nandi et al. [49] believe that the COVID-19 pandemic has caused enterprises and society to face the insufficiency of normal production and consumption patterns. In order to make the supply chain more flexible, transparent, and sustainable [49] proposed potential solutions that use blockchain technology and circular economy principles to lay the foundation for future research on sustainable production and consumption.

6. Conclusions

In the business environment, in order to recover value and increase the company's profitability, product returns and product recycling in reverse logistics have become very important. Today's information system is developed based on the traditional logistics process without considering the specific characteristics of reverse logistics [50]. The ESPRES system proposed in this study adopted blockchain technology to keep all transaction footprints in the large home appliances delivery and return network reliable and safely. The characteristics of blockchain technology created a platform of reverse logistics that is trustworthy without a third-party trust. It can also prevent return fraud in reverse logistics in any area. Nevertheless, adding PROMETHEE for freight fee decision support in smart contracts not only optimizes users' costs but also reduces possible freight disputes in the future. A practical implication of this study is that by adopting blockchain technology, the manpower to check whether each return or exchange is fraudulent can be reduced. Merchants can check product ownership and stored invoice information through smart contracts for information verification to prevent fraud. In addition, due to the fact that the footprint of goods cannot be forged, this study can prevent counterfeit or parallel imports of goods, improve the credibility of the company, optimize customer costs, and even create sales for the company.

The current limitation of this study is that the Ethereum environment can only use a single role for activities, and we cannot choose another role for other activities, making it difficult to implement. For future works, we consider using different blockchain implementation environments and comparing their differences. In addition, due to the rise of smart life and smart home appliances, the Internet of Things can be added to the blockchain implementation system as a future study on the feasibility.

Author Contributions: Conceptualization, D.-H.S. and F.-C.H.; Data curation, C.-Y.C.; Formal analysis, M.-H.S.; Funding acquisition, D.-H.S.; Investigation, F.-C.H.; Methodology, D.-H.S.; Project administration, D.-H.S. and T.-W.W.; Resources, C.-Y.C.; Supervision, F.-C.H. and T.-W.W.; Validation, M.-H.S.; Writing—original draft, C.-Y.C.; Writing—review & editing, M.-H.S. and T.-W.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Taiwan Ministry of Science and Technology grant MOST 109-2410-H-224-022 and MOST 110-2410-H-224-010. The funder has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Verhoef, P.C.; Kannan, P.K.; Inman, J.J. From multi-channel retailing to omni-channel retailing: Introduction to the special issue on multi-channel retailing. *J. Retail.* **2015**, *91*, 174–181. [CrossRef]
2. Reitblat, M. Returns Abuse: A $24 Billion Problem, TotalRetail. 2020. Available online: https://www.mytotalretail.com/article/returns-abuse-a-24-billion-problem/ (accessed on 14 April 2021).
3. Daniel Parisi, Return Fraud Represents $2.2 Billion in Lost Sales—Omnichannel Retailers Are Particularly Vulnerable. December 2015. Available online: https://geomarketing.com/return-fraud-represents-2-2-billion-in-lost-sales-omnichannel-retailers-are-particularly-vulnerable (accessed on 25 May 2021).
4. Schmidt, R.A.; Sturrock, F.; Ward, P.; Lea-Greenwood, G. Deshopping—The art of illicit consumption. *Int. J. Retail Distrib. Manag.* **1999**, *27*, 290–301. [CrossRef]
5. King, T.; Dennis, C.; McHendry, J. The management of deshopping and its effects on service: A mass market case study. *Int. J. Retail Distrib. Manag.* **2007**, *35*, 720–733. [CrossRef]
6. Speights, D.; Hilinski, M. Return fraud and abuse: How to protect profits. *Retail. Issues Lett.* **2005**, *17*, 1–6.
7. Zhuravlev, Y.; Dokukin, A.; Senko, O.; Stefanovskiy, D. Use of Clusterization Technique to Highlight Groups of Related Goods by Digital Traces in Retail Trade. In Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 5–7 June 2019; pp. 84–88.
8. Pagés, R.; Amplianitis, K.; Monaghan, D.; Ondřej, J.; Smolić, A. Affordable content creation for free-viewpoint video and VR/AR applications. *J. Vis. Commun. Image Represent.* **2018**, *53*, 192–201. [CrossRef]
9. Henderson, R. Using graph databases to detect financial fraud. *Comput. Fraud Secur.* **2020**, *2020*, 6–10. [CrossRef]
10. Del Mar Roldán-García, M.; García-Nieto, J.; Aldana-Montes, J.F. Enhancing semantic consistency in anti-fraud rule-based expert systems. *Expert Syst. Appl.* **2017**, *90*, 332–343. [CrossRef]
11. Toyoda, K.; Mathiopoulos, P.T.; Sasase, I.; Ohtsuki, T. A novel blockchain-based product ownership management system (POMS) for anti-counterfeits in the post supply chain. *IEEE Access* **2017**, *5*, 17465–17477. [CrossRef]
12. Bower, A.B.; Maxham, J.G. Return shipping policies of online retailers: Normative assumptions and the long-term consequences of fee and free returns. *J. Market.* **2012**, *76*, 110–124. [CrossRef]
13. Vitasek, K. *Supply Chain Management Terms and Glossary*; Supply Chain Visions: Bellevue, WA, USA, 2009.
14. Fleischmann, M. *Quantitative Models for Reverse Logistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2001; Volume 501.
15. Rubio, S.; Jiménez-Parra, B. Reverse logistics: Overview and challenges for supply chain management. *Int. J. Eng. Bus. Manag.* **2014**, *6*, 12. [CrossRef]
16. Prajapati, H.; Kant, R.; Shankar, R. Bequeath life to death: State-of-art review on reverse logistics. *J. Clean. Prod.* **2019**, *211*, 503–520. [CrossRef]
17. Wilkes, R.E. Fraudulent Behavior by Consumers: The other Side of Fraud in the Marketplace: Consumer-Initiated Fraud against Business. *J. Market.* **1978**, *42*, 67–75. [CrossRef]
18. King, T. To Examine the Phenomenon of Deshopping and Retail Policies Preventing Deshopping. Master's Thesis, Manchester Metropolitan University, Manchester, UK, 1999.
19. King, T. An analysis of the Phenomenon of Deshopping of Garments in Women's Wear Retailing. Ph.D. Thesis, Brunnel University, Brunel, UK, 2004.
20. Jendruszak, B. Ecommerce's Dirty Secret: The Growing Problem of Return Fraud and How to Fight It. Available online: https://seon.io/resources/how-to-fight-return-fraud/ (accessed on 5 May 2021).
21. Li, J.; Wang, X. Research on the Application of Blockchain in the Traceability System of Agricultural Products. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 2637–2640.
22. Lin, I.C.; Liao, T.C. A Survey of Blockchain Security Issues and Challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659.
23. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: Bitcoin.org/bitcoin.pdf (accessed on 5 May 2021).
24. Wood, G. Ethereum: A secure decentralized generalized transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
25. Sunny, J.; Undralla, N.; Pillai, V.M. Supply chain transparency through blockchain-based traceability: An overview with demonstration. *Comput. Ind. Eng.* **2020**, *9*, 106895. [CrossRef]
26. Bhatt, P.C.; Kumar, V.; Lu, T.C.; Daim, T. Technology convergence assessment: Case of blockchain within the IR 4.0 platform. *Technol. Soc.* **2021**, *67*, 101709. [CrossRef]
27. Kassen, M. Blockchain and e-government innovation: Automation of public information processes. *Inf. Syst.* **2021**, *103*, 101862. [CrossRef]
28. Gong, Y.; van Engelenburg, S.; Janssen, M. A Reference Architecture for Blockchain-Based Crowdsourcing Platforms. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 937–958. [CrossRef]
29. Wessling, F.; Ehmke, C.; Hesenius, M.; Gruhn, V. How much blockchain do you need? Towards a concept for building hybrid dapp architectures. In Proceedings of the 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Gothenburg, Sweden, 27 May–3 June 2018; pp. 44–47.

30. Wüst, K.; Gervais, A. Do you need a Blockchain? In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), Zug, Switzerland, 20–22 June 2018; pp. 45–54.
31. Luhmann, N. *Trust and Power*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
32. Brans, J.P.; Mareschal, B. Promethee Methods. In *Multiple Criteria Decision Analysis: State of the Art Surveys*; International Series in Operations Research & Management Science; Springer: New York, NY, USA, 2005; Volume 78. [CrossRef]
33. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
34. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 254–269.
35. Hong, G.; Yang, Z.; Yang, S.; Lei, Z.; Nan, Y.; Zhang, Z.; Duan, H. How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 1701–1713.
36. Li, H.; Han, D. EduRSS: A Blockchain-Based Educational Records Secure Storage and Sharing Scheme. *IEEE Access* **2019**, *7*, 179273–179289. [CrossRef]
37. Tsankov, P.; Dan, A.; Drachsler-Cohen, D.; Gervais, A.; Buenzli, F.; Vechev, M. Securify: Practical security analysis of smart contracts. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 67–82.
38. Chen, L. Research of E-commerce Supply Chain Management with CreditRisk+ Model. In Proceedings of the 2012 International Conference on Management of e-Commerce and e-Government, Beijing, China, 20–21 October 2012; pp. 353–355.
39. Xu, L.; Chen, L.; Gao, Z.; Lu, Y.; Shi, W. Coc: Secure supply chain management system based on public ledger. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–6.
40. Kostyuk, P.; Kudryashov, S.; Madhwal, Y.; Maslov, I.; Tkachenko, V.; Yanovich, Y. Blockchain-Based Solution to Prevent Plastic Pipes Fraud. In Proceedings of the 2020 Seventh International Conference on Software Defined Systems (SDS), Paris, France, 20–23 April 2020; pp. 208–213.
41. Cueva-Sánchez, J.J.; Coyco-Ordemar, A.J.; Ugarte, W. A blockchain-based technological solution to ensure data transparency of the wood supply chain. In Proceedings of the 2020 IEEE ANDESCON, Quito, Ecuador, 13–16 October 2020; pp. 1–6.
42. Cai, Y.; Zhu, D. Fraud detections for online businesses: A perspective from blockchain technology. *Financ. Innov.* **2016**, *2*, 1–10. [CrossRef]
43. Akın, Y.; Dikkollu, C.; Kaplan, B.B.; Yayan, U.; Yolaçan, E.N. Ethereum Blockchain Network-based Electrical Vehicle Charging Platform with Multi-Criteria Decision Support System. In Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, 6–7 November 2019; pp. 1–5.
44. Farida, Y.; Khasanah, Z.S.U. Analisis Performa Mata Uang Virtual (Cryptocurrency) Menggunakan Preference Ranking Organization Method for Enrichment Evaluation (Promethee). *Rekayasa* **2021**, *14*, 1–9. [CrossRef]
45. Sarkis, J. Supply chain sustainability: Learning from the COVID-19 pandemic. *Int. J. Oper. Prod. Manag.* **2020**, *41*, 63–73. [CrossRef]
46. Singh, M.; Aujla, G.S.; Bali, R.S.; Vashisht, S.; Singh, A.; Jindal, A. Blockchain-enabled secure communication for drone delivery: A case study in COVID-like scenarios. In Proceedings of the 2nd ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and beyond, London, UK, 25 September 2020; pp. 25–30.
47. Siriwardhana, Y.; De Alwis, C.; Gür, G.; Ylianttila, M.; Liyanage, M. The fight against the COVID-19 pandemic with 5G technologies. *IEEE Eng. Manag. Rev.* **2020**, *48*, 72–84. [CrossRef]
48. Kumar, M.S.; Raut, R.D.; Narwane, V.S.; Narkhede, B.E. Applications of industry 4.0 to overcome the COVID-19 operational challenges. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2020**, *14*, 1283–1289. [CrossRef] [PubMed]
49. Nandi, S.; Sarkis, J.; Hervani, A.A.; Helms, M.M. Redesigning supply chains using blockchain-enabled circular economy and COVID-19 experiences. *Sustain. Prod. Consum.* **2021**, *27*, 10–22. [CrossRef]
50. Oltra-Badenes, R.; Gil-Gomez, H.; Guerola-Navarro, V.; Vicedo, P. Is It Possible to Manage the Product Recovery Processes in an ERP? Analysis of Functional Needs. *Sustainability* **2019**, *11*, 4380. [CrossRef]